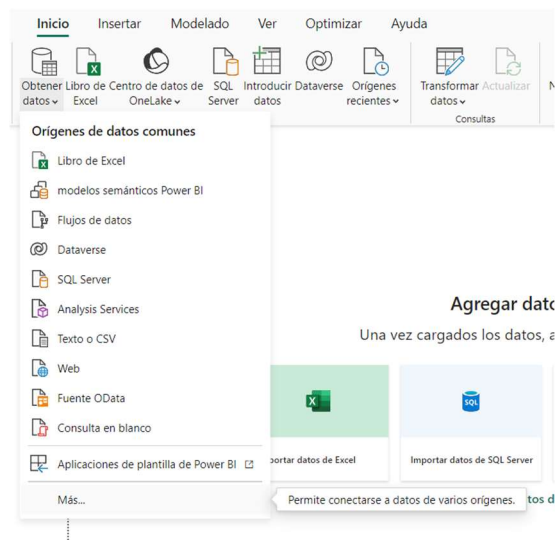


**SPRINT 8.2**  
**Tasca S8.02. Power BI amb Python**  
**HAILE JACOBO MENESES MORENO**

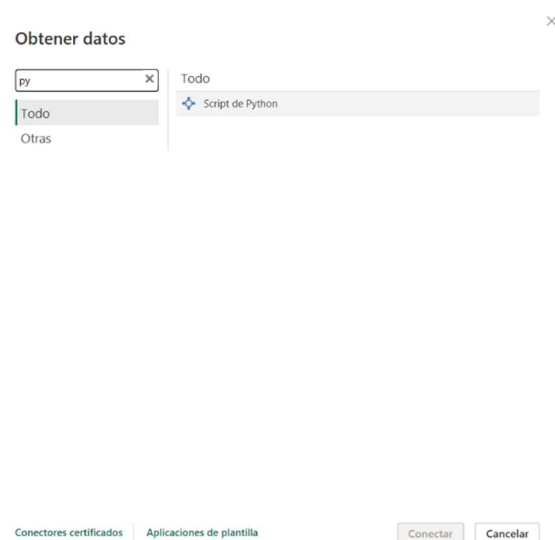
**NIVEL 1**

**PRIMER PASO. Conexión de Python con Power BI**

Hago click en el botón de Obtener Datos para que aparezca el listado de Orígenes de Datos Comunes:



En la ventana de búsqueda comienzo a escribir Python para que el texto predictivo localice rápidamente la opción Script de Python



En el Cuadro de Dialogo del Script introduzco el script de conexión utilizado en el ejercicio del Sprint 8.1 para crear la conexión y generar los dataframes. Como comentario adicional sobre el resultado esperado para la ejecución de los scripts dentro de Power BI, incluí la importación de las librerías para ver si sería suficiente para cargarlas una sola vez, no funcionó y tuve que cargarlas con cada script.

### Script de Python

Script

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sqlalchemy import create_engine

conexion = create_engine(f'mysql+mysqlconnector://root:patitos26@localhost/transactions4')

tablas_bbdd = ['company', 'comprados', 'credit_card', 'products', 'transactions', 'users']
df = {}

for tabla in tablas_bbdd:
    query = f'SELECT * FROM {tabla}'
```

El script se ejecutará con la instalación de Python siguiente:

C:\Users\menes\AppData\Local\Programs\Python\Python312.

Para establecer la configuración y cambiar la instalación de Python que quiere ejecutar, vaya a Opciones y configuración.

Aceptar

Cancelar

### Script de Python

Script

```
for tabla in tablas_bbdd:
    query = f'SELECT * FROM {tabla}'
    df[tabla] = pd.read_sql(query, conexion)

conexion = conexion.dispose()

df_company = df['company']
df_comprados = df['comprados']
df_credit_card = df['credit_card']
df_products = df['products']
df_transactions = df['transactions']
df_users = df['users']
```

El script se ejecutará con la instalación de Python siguiente:

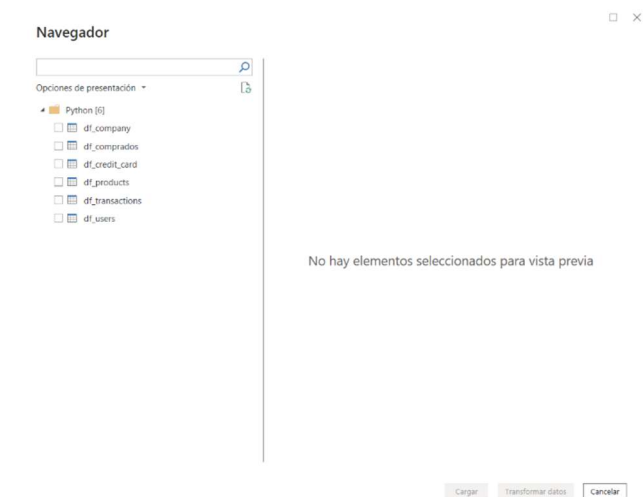
C:\Users\menes\AppData\Local\Programs\Python\Python312.

Para establecer la configuración y cambiar la instalación de Python que quiere ejecutar, vaya a Opciones y configuración.

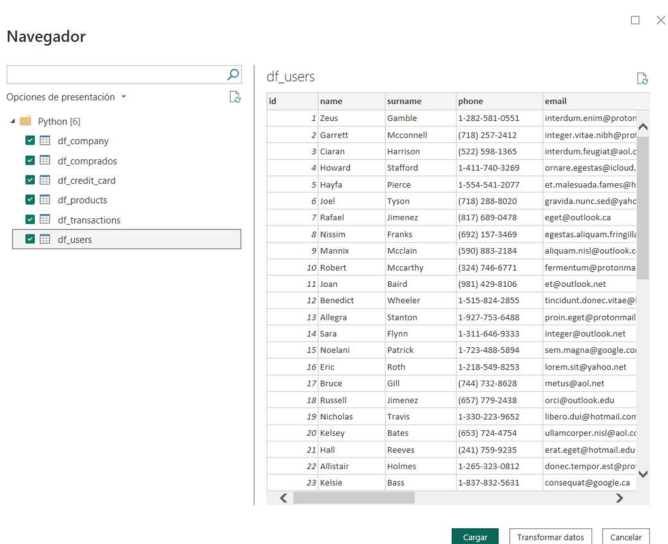
Aceptar

Cancelar

Al dar click en Aceptar se abre el siguiente cuadro de dialogo.

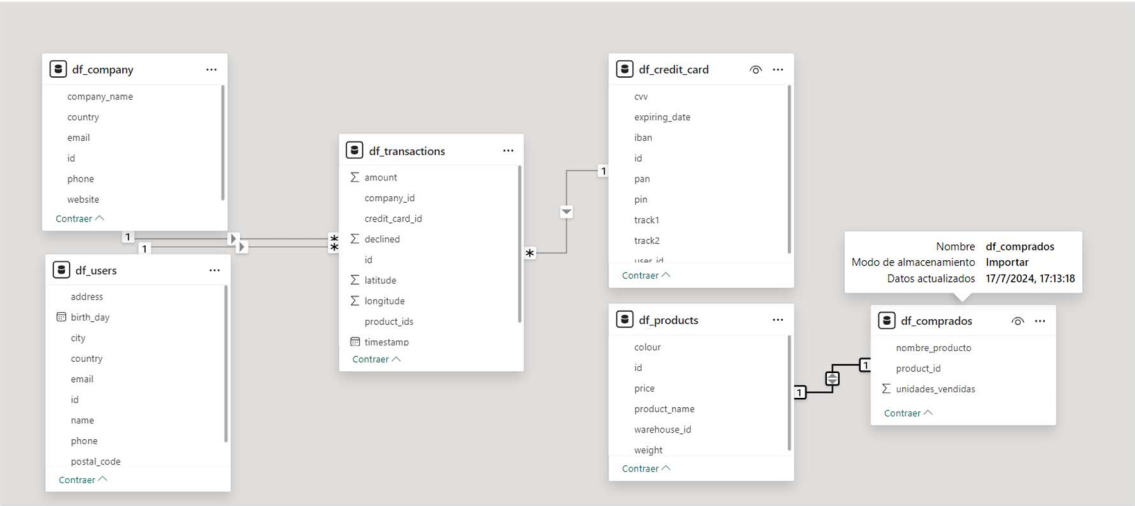


Selecciono todos los dataframes donde se pueden ver los datos a cargar:



SEGUNDO PASO. Establecer las relaciones.

Una vez cargados los datos (dataframes) me dispongo a establecer las relaciones para así poder trabajar eficientemente los scripts con Power BI al momento de utilizar las referencias entre tablas(dataframes). Aquí el resultado de las relaciones establecidas:



TERCER PASO. Transformación de los Datos.

Posteriormente selecciono Transformar Datos para ver si los datos han sido cargados correctamente, observo que tanto “price” como “weight” han sido pasado como tipo texto y con punto; la configuración de Power BI trabaja con comas y debo además de corregir este detalle para trabajar con los datos.

= Origen([Name="df_products"])[Value]					
A <sup>0</sup> id	A <sup>0</sup> product_name	A <sup>0</sup> price	A <sup>0</sup> colour	A <sup>0</sup> weight	A <sup>0</sup> warehouse_id
1	Direwolf Stannis	161.11	#7c7c7c	1	WH-4
2	Karstark Dorne	119.52	#f4f4f4	2.4	WH-5
3	south duel	40.43	#6d6d6d	3	WH-95
4	Karstark Dorne	49.70	#141414	2.7	WH-6
5	duel Direwolf	181.60	#a8a8a8	2.1	WH-7
6	palpatine chewbacca	139.59	#2b2b2b	1	WH-8
7	Direwolf	147.53	#c4c4c4	2	WH-9
8	Stannis warden	194.29	#bdbdbd	1.5	WH-10
9	the duel warden	180.91	#666666	3	WH-11
10	skywalker ewok sith	91.89	#7c7c7c	3.2	WH-12
11	Karstark warden	148.91	#c4c4c4	0.8	WH-13
12	dooku solo	60.33	#f3f3f3	0.6	WH-14
13	Tarly Stark	9.24	#919191	2	WH-3
14	warden Karstark	91.96	#b5b5b5	1.4	WH-15
15	duel Direwolf	96.90	#e2e2e2	1.2	WH-16
16	chewbacca mustafar	150.02	#fcfcfc	2.4	WH-17
17	riverlands north	169.96	#545454	2.7	WH-18
18	south duel tourney	48.99	#aaaaaa	2.1	WH-19
19	skywalker ewok	157.53	#2b2b2b	1	WH-20
20	Stark Karstark	53.01	#989898	2	WH-21
21	Stannis riverlands	172.93	#7a7a7a	1.5	WH-22
22	chewbacca mustafar	127.44	#efefef	3	WH-23
23	Tully maester Tarly	167.20	#111111	3.2	WH-24
24	duel tourney Lannister	171.13	#8d8d8d	1.5	WH-2
25	Karstark warden	79.53	#606060	0.8	WH-25
26	Lannister	85.02	#f3f3f3	0.6	WH-26
27	north	178.28	#0f0f0f	1.4	WH-27
28	duel warden	127.09	#191919	1.2	WH-28

Como primer paso reemplazo los puntos por comas para poder realizar el cambio de tipo de dato eficientemente, sin este paso previo no trata los datos correctamente:

Reemplazar los valores

Reemplace un valor con otro de las columnas seleccionadas.

Valor que buscar

Reemplazar con

,

Opciones avanzadas

Aceptar

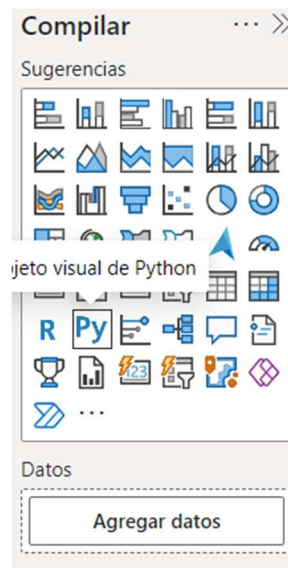
Cancelar

Una vez hecho esto cambio el tipo de dato:

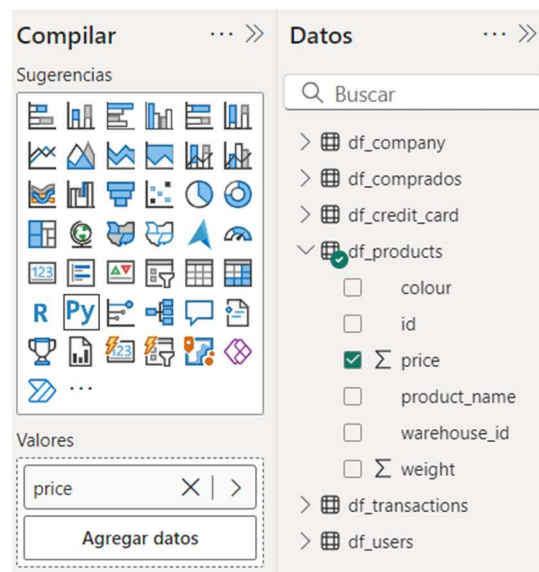
Table.TransformColumnTypes(#"Valor reemplazado1",{{"price", type number}, {"weight", type number}})					
id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	161,11	#7c7c7c		1 WH-4
2	Karstark Dorne	119,52	#f4f4f4		2,4 WH-5
3	100 south duel	40,43	#d6d6d6		3 WH-95
4	11 Karstark Dorne	49,7	#141414		2,7 WH-6
5	12 duel Direwolf	181,6	#a8a8a8		2,1 WH-7
6	13 palpatine chewbacca	139,59	#2b2b2b		1 WH-8
7	14 Direwolf	147,53	#c4c4c4		2 WH-9
8	15 Stannis warden	194,29	#bdbdbd		1,5 WH-10
9	16 the duel warden	180,91	#666666		3 WH-11
10	17 skywalker ewok sith	91,89	#7c7c7c		3,2 WH-12
11	18 Karstark warden	148,91	#c4c4c4		0,8 WH-13
12	19 dooku solo	60,39	#f3f3f3		0,6 WH-14
13	2 Tarly Stark	9,24	#919191		2 WH-3
14	20 warden Karstark	91,96	#b5b5b5		1,4 WH-15
15	21 duel Direwolf	96,9	#e2e2e2		1,2 WH-16
16	22 chewbacca mustafar	150,02	#fcfcfc		2,4 WH-17
17	23 riverlands north	169,96	#545454		2,7 WH-18
18	24 south duel tourney	48,99	#aaaaaa		2,1 WH-19
19	25 skywalker ewok	157,53	#2b2b2b		1 WH-20
20	26 Stark Karstark	53,01	#898989		2 WH-21
21	27 Stannis riverlands	172,93	#7a7a7a		1,5 WH-22
22	28 chewbacca mustafar	127,44	#efefef		3 WH-23
23	29 Tully maester Tarly	167,2	#111111		3,2 WH-24
24	3 duel tourney Lannister	171,13	#d8d8d8		1,5 WH-2

#### CUARTO PASO. Creación de los Gráficos

Selecciono Objeto Visual de Python:



Selecciono los datos (variables) que necesitaré en el objeto/script. En este caso uso el del primero ejercicio "Price" de el DF products:



El Script original tiene en los parámetros de origen de dato del gráfico la ruta `df_products['price']`:

```
1 # Ejercicio 1.
2
3 sns.histplot(data=df_products['price'], kde=True, bins=10)
4 plt.title('Histograma de Precio')
5 plt.xlabel('Precio')
6 plt.ylabel('Cantidad')
7 plt.show()
```

Pero en PowerBI tal como se observa en el comentario al inicio predeterminado del script nos indica que el origen debe ser el dataset sin especificar la columna, pues esta la hemos seleccionado al principio, Power BI ya establece esta relación y nos hace ya el trabajo:

```
# dataset = pandas.DataFrame(price)
# dataset = dataset.drop_duplicates()

# Pegue o escriba aquí el código de script:
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize= (10,6))
sns.histplot(data=dataset, kde=True, bins=10)
plt.title('Histograma de Precio')
plt.xlabel('Precio')
plt.ylabel('Cantidad')
plt.show()
```

Como detalle adicional tenemos que indicar en el script al principio el tamaño del gráfico con el parámetro `plt.figure`, probando varias opciones la proporción (10,6) ha sido la que ha permitido obtener la mejor opción en todos los gráficos.

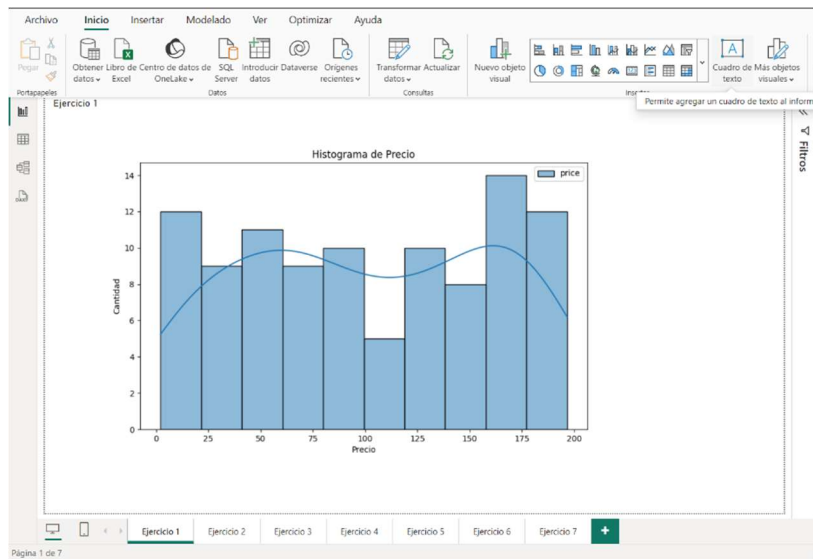
Adicionalmente y como mencioné al principio, es necesario importar en todos los ejercicios incluir la importación de las librerías que se han de utilizar:

```
# dataset = pandas.DataFrame(price)
# dataset = dataset.drop_duplicates()

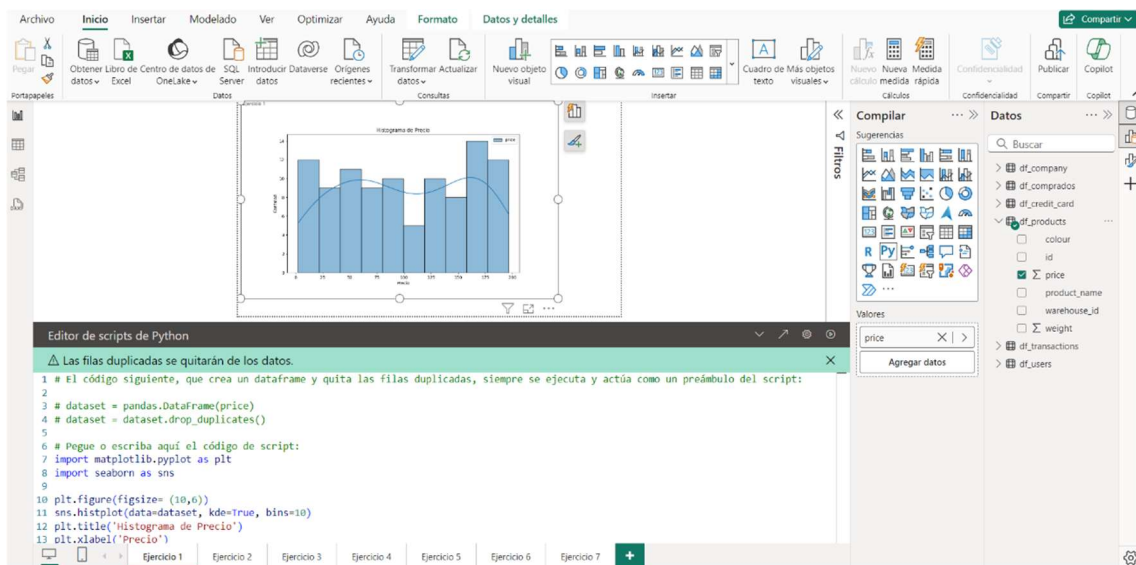
# Pegue o escriba aquí el código de script:
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize= (10,6))
sns.histplot(data=dataset, kde=True, bins=10)
plt.title('Histograma de Precio')
plt.xlabel('Precio')
plt.ylabel('Cantidad')
plt.show()
```

El gráfico resultante es el siguiente:



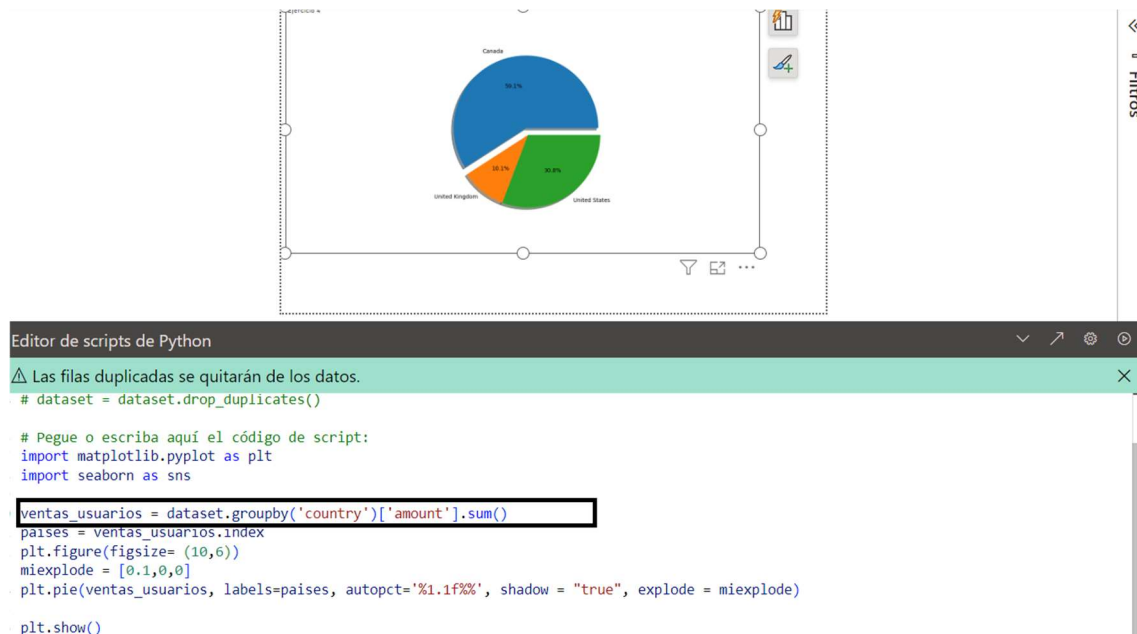
Todo el procedimiento queda establecido finalmente de esta manera:



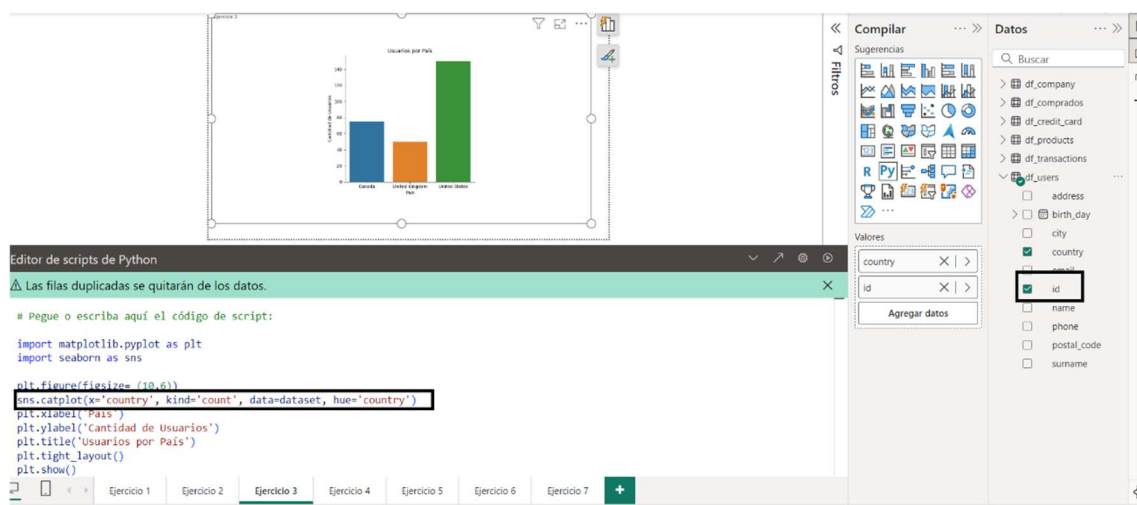


## Resto Ejercicios.

El procedimiento se repite de manera similar en todos los ejercicios, en algún caso como en el siguiente gráfico (Ejercicio 4) se modifica la referencia del “df\_users” por “dataset” para que el script funcione correctamente, como lo he mencionado anteriormente: Power BI hace parte del trabajo:



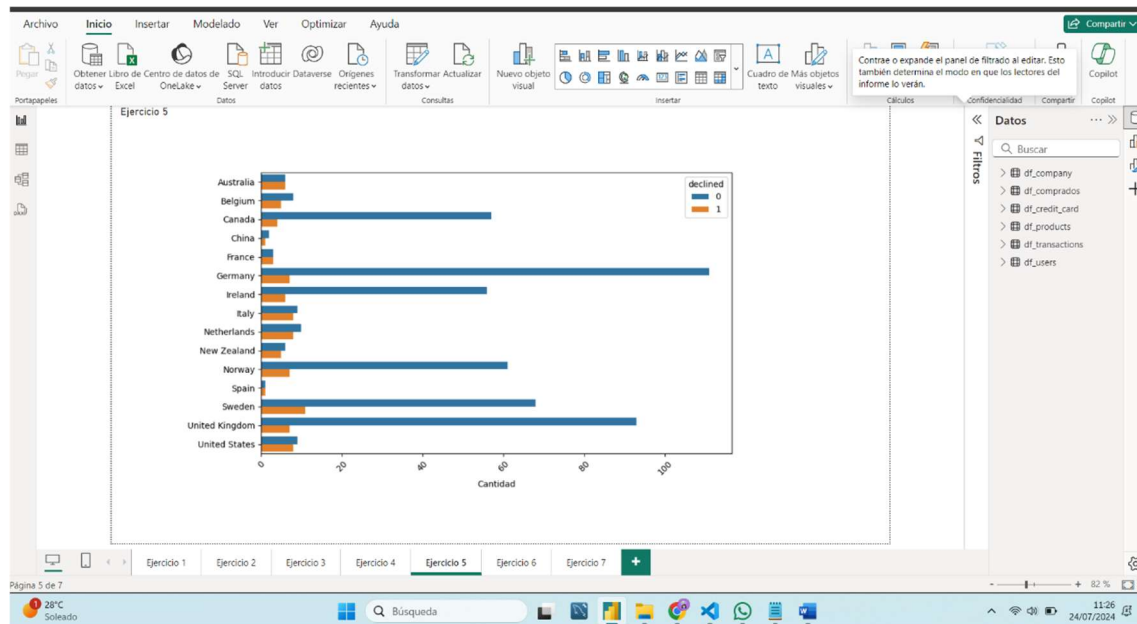
Otro detalle que se ha de considerar, por ejemplo en este gráfico (Ejercicio 3) es que para que el parámetro kind='count' funcione correctamente, debe seleccionarse la columna “id” del DF que corresponda en el gráfico (en este caso del df\_users)



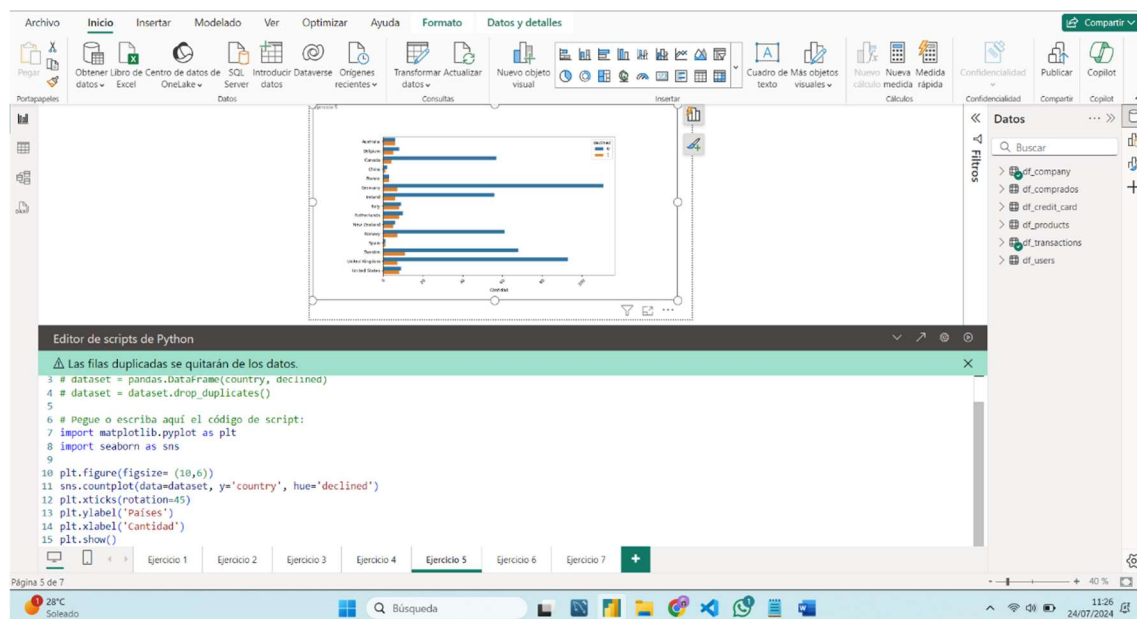
## ANEXO SOBRE LO SOLICITADO EN LA REVISIÓN DEL SPRINT 24/07/2024

### Nivel 1. Ejercicio 5

Cargando el archivo nuevamente adjunto las capturas de pantalla y el script Python correspondiente:



Captura del gráfico solicitado observándose que se ha cargado correctamente (ver fecha y hora para contrastar dato actual si es necesario)



Captura del gráfico con el script que lo ejecuta; fecha y hora para referencia.

En el gráfico del Nivel 1. Ejercicio 3 adjunto pantallazo que la proporción del gráfico es el mismo que en 8.1 (fecha y hora de referencia). No tengo explicación de que se cargue de manera distinta en el Power BI del ordenador de revisión.

